Integration of distributed spatial data in a decision making context

Christelle Pierkot

Integration of distributed spatial data in a decision making context

Christelle Pierkot

LIRMM, D'OC Team

161 rue ADA

34392 Montpellier Cedex 5, France

christelle.pierkot@lirmm.fr

Abstract. The widespread deployment of communication networks has facilitated the sharing and exchange of spatial data between producers and users spread out over different locations. These users are not necessarily located at the same sites neither belong to the same organizations, but must cooperate. Spatial data is used to help user making decision in this context. Data is updated in parallel, depending on the requirements and, consequently, at different levels of detail and quality. Problems then arise during the integration of evolutions originating from multiple sources into the final user's dataset. To solve these problems, we propose an overall process of integrating evolutions that will, on the one hand, allow the filtering out of evolutions that are irrelevant to the end user and, on the other, to detect and process conflicts caused by the updating.

Keywords Decision making, Evolutions, Integration, Consistency, Relevancy, Reconciliation, Metadata

1 Introduction

The widespread deployment of communication networks has facilitated the sharing and exchange of spatial data between producers and users spread out over different locations.
For example, the Languedoc Roussillon regional authority wants to constitute an observatory to support the sustainable and integrated management of its coastal area. This implies that experts in various fields must work together and take decisions that will impact on the future observatory. These users are not necessarily located at the same sites neither belong to the same organizations, but must cooperate. Spatial data is used to help user making decision in this context. Thus, the information is not centralized, data is distributed and also might be replicated at each site and users could periodically be disconnected from the network, for example, while a mobile user takes field measurements. However, information (data and updates) must be regularly exchanged to ensure a good collaboration between all the parts involved in this program

In this context, data is updated in parallel, depending on the requirements and, consequently, at different levels of detail and quality. Nevertheless, multidirectional exchanges have to take place to ensure cooperation and coordination between the various actors. Problems then arise during the integration of evolutions originating from multiple sources into the final user's dataset. In fact, the updates received can even be in conflict with the user dataset and thus lead to inconsistencies. Moreover, they are not necessarily relevant to him. These two problems can cause a lot of troubles in a decision making process.

To solve these problems, we propose an overall process of integrating evolutions that will, on the one hand, allow the filtering out of evolutions that are irrelevant to the end user and, on the other, to detect and process conflicts caused by the updating. Thus, we propose a mechanism to detect conflicting updates linked to the data types handled and some procedures to reconcile diverging writes which are best suited to the final user.

This strategy relies on the prior establishment of a spatial data infrastructure [11],[13] in which a communications network is defined, the users and their different roles are known, the useful metadata has been specified and wherein the exchange of updates can be conducted according to a common strategy. To this end, we have defined a model that allows the specifying of the links between all three entities involved in the infrastructure : the data, the actors and the evolutions; and also defined a metadata profile conforming to the ISO 19115 standard to help manage the evolutions in a decision making context.

This strategy is therefore original in the way it relies on standardized metadata to provide reconciliation solutions by taking into account the needs of the user.

In this paper, we will first outline the problems of consistency of replicated data. Then we will present the strategy that we have defined for integrating multi-sources evolutions. We explain the general approach of the process by describing each of the stages we have implemented and in section 4, we present the metadata model used in the strategy. Finally, we analyse the obtained results and conclude.

2 Consistency's problems of replicated data in a decision making context

Optimistic replication is a very active field of research in database and systems studies [3] [10]12] [15].

The main characteristics of an optimistic replication system are [14]:

- Each replication site has a copy of shared objects which it can modify freely at any time.
- The update is executed immediately locally, then sent to the other sites for later execution.
- All writes are accepted a priori, which means that the updates are potentially concurrent and can be the source of inconsistencies.

- The resolution is effected a posteriori during the synchronization, which detects possible conflicts.

In optimistic replication, there is consistency when the system converges to a common final state, i.e., when the copies become identical at the end of the propagation of all operations on every site. However, in geographic information, the consistency is ensured when the data produced does not represent an absurd view of the real world [2]. This means that conflicts that may lead to inconsistencies during integration into the target dataset have to be defined and processed. Three main types of conflicts are generally mentioned in the literature: model conflicts (data-representation model, geometric model, topological model, etc.), structural conflicts (grouping, generalization, etc.) and semantic conflicts (interpretation depending on local context) [1]. In our study, we assume that model and structural conflicts are handled further upstream and we limit ourselves to handling semantic conflicts, which can be divided into two categories: spatial conflicts (topological and geometric) and thematic conflicts (attributes). We'll see that the convergence, in the sense of equivalence of copies, is not systematically ensured.

The problem of concurrence of replicated data has been studied by the systems [3] [7] and database [12] [15] communities. Protocols have been defined but most of them, assume the existence of a reference server for data centralization. The mechanisms used in geographic information are the 'check-in, check-out' approach [4] and versioning [7]. In the first case, other users are locked out from the data until the updates are integrated, the second requires a centralized server to hold reference data. Neither of these methods is therefore fully suitable for our distributed context where the data is replicated on every site, can evolve in parallel and differently, and has to be available to users.

In addition, in the framework of decision making, different consistency levels (strong, weak, intermediate) are desired depending on the type, the localization, the role, the goals and the requirements of the users. Indeed, users which handle these spatial data have different roles

- Producers must supply reference information and prepare future datasets. They use substantial hardware resources to update their data and then to share it with the other users. The goal here is to gather quality information to be able to obtain a reliable and precise dataset. An effort is made to minimize the appearance of inconsistencies during the integration of evolutions into the producer's dataset. The consistency level is therefore considered strong. Consistency constraints which allow the resolution of conflicts are defined from the very beginning of the mission. They do not depend on the context, nor can be changed. This ensures the convergence between producers, in the sense of equivalence of copies.
- In our context, two different types of users are considered :
    - The simple users' role is to consult data to be able to make decisions. They cannot update their data, nor share it – but can receive it from other users. The goal here is to gather a maximum amount of information relevant to a particular requirement, irrespective of its quality. We accept here that inconsistencies will appear when evolutions

are integrated into the user dataset. The consistency level desired can be considered weak. Consistency constraints are defined depending on the context (user requirements, urgency of the situation, work to be done, etc.) and can change over the time. The convergence is therefore not ensured here.

- The operational users (mobile or not) have a dual role: rapidly supply updates to local users and to send back the information entered to the producers. These users only have simple hardware resources to update their data and then to share it with the other actors. The goal here is to supply a maximum amount of information while retaining a certain level of quality. We try to limit but nevertheless accept some inconsistencies during the integration of evolutions into the dataset. The consistency level aimed for is therefore intermediate. The consistency constraints depend on the context and can change. Convergence is therefore not ensured.

Therefore, in such a context, we have to manage the consistency at different levels of a geographic dataset's different replicas during the integration of evolutions originating from different sources. Convergence, in the strictest sense of the term, is ensured only for the producers.

3 Integration strategy of multi-sources evolutions

To resolve the problem of consistency of replicated spatial data, we propose an overall integration process of evolutions whose objective is to facilitate integration of a continuous stream of evolutions originating from multiple sources into a particular dataset. This strategy does not aim for a strict convergence for all the users. Here, we explain first the general approach of the strategy by describing each of the stages we have defined. Then, we show how these stages are organized into the overall strategy.

3.1. General approach

Figure 1 shows the general approach of the integration strategy. It consists of three stages and can be applied to a user dataset located at different sites, for which evolutions originating from different sources are destined. For updating the user's dataset, it is necessary to execute all three stages of the integration strategy.

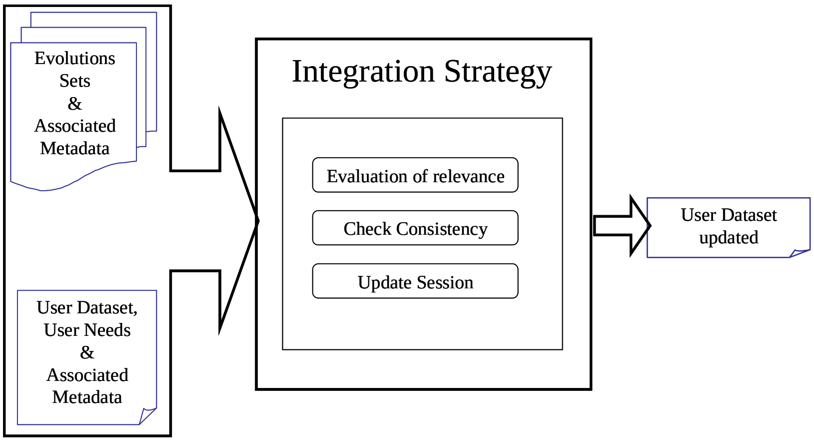

Figure 1: Integration strategy of updates into a user dataset

The first stage consists of the evaluation of the relevance of the evolutions. The aim of a process to verify the relevance is to filter out, from amongst all the evolutions, those that are not relevant to the user and which, if integrated, would risk impairing the external quality[1] of the user's dataset. The solution that we recommend for this filtering relies on the matching of metadata associated with evolutions and with the users' needs and is based on the work of Jeansoulin [9] and Vasseur [16]. We draw inspiration from this work for evaluating the external quality of a set of evolutions with respect to the usage the end user could put it to. At the end of this stage, the set of evolutions proposed for the integration only contains those evolutions that are relevant to the target user. Finally, this stage reduces the sets of evolutions to be integrated to a restricted and relevant number of evolutions.

The second stage concerns consistency checking. The goal is to detect and process any possible conflicts that could result from the multiple origins of the sets of evolutions. We have seen that we have to manage the consistency at several levels, levels which we determine based on the role and objectives of the actors into the infrastructure. We therefore propose a two-phase protocol that allows, on the one hand, the detection of conflicts which can lead to inconsistencies and, on the other, offers routines for the reconciliation of conflicting evolutions as a function of the desired level of consistency. In addition, we have to verify the consistency between the actor's data and the proposed evolutions and between all the evolutions which are candidates for integration. In fact, within the infrastructure, several actors are responsible for updating datasets and an evolution of a real-world phenomenon could have been entered several times, at different

---

[1] The external quality is defined as being the suitability of specifications to the user's requirements.

locations or at different times. These evolutions and this data can then clash with each other and thus lead to inconsistencies.

We have specified three types of conflicts: *update conflicts* (modification or deletion of an existing object), *topological conflicts* (intersection or overlapping of objects) and the *conflicts of creation* (multiple creations of a same object). We use the structure of the evolutions defined in the infrastructure (object identifiers) and geometrical processes (tests on the spatial relationships between objects and matching techniques) to then detect possible conflicts during the first phase of the protocol.

The reconciliation protocol (second phase of the protocol) is called upon when one or several conflicts have been detected during the previous phase. This process's objective is to provide a solution suitable for resolving the conflict(s). The result depends on the desired consistency level and the desired balance between quality and quantity. The reconciliation protocol we propose is original in its use of the metadata associated with the entities of the DAE model to offer a result commensurate with the actors' expectations. Metadata included with the evolutions and the data provides information such as on its quality and its origin[2]. This information allows the process to compare the items with the user's expectations. Thus, when a conflict has to be processed, the process can make a choice and propose a reconciliation which depends on the desired consistency level and usage. The reconciliation process consists of several stages:

- First, a comparison of metadata associated with the items in conflict is made with the actors' metadata and a calculation for measuring the quality specific to each of the item's characteristics (geometry, attributes, reliability, etc.) is made. This part is based on the work done on the calculation of utility by Grum and Frank [5] [6].
- Then, we calculate a measurement of overall quality for each item in conflict to be able to obtain a result that is a function of the final user's expectations and of the desired consistency level. This part is based on the work of Vasseur [16].
- Finally, a comparison of the overall quality measurements of the items in conflict is made and the item found to be the most relevant is chosen for future integration. This part can be executed automatically, semi-automatically or interactively.

At the end of this stage, we obtain a set consisting of relevant and non-conflicting evolutions that we can integrate into the user's dataset.

The third and final stage allows the integration of the evolutions that have already been processed into the user's dataset. We use update sessions for two main reasons: to limit the actions on the user's data and to obtain persistent states of the user's dataset. In fact, since multi-source evolutions arrive in a continuous stream, several updates concerning the same object can be received. If integration was done immediately, i.e., as and when an update which has already been processed was received, one would have to undo and redo the

[2] The metadata model is explained in more detail in section 4

actions, with attendant problems of repositioning. Using sessions also allows us to have dataset states that can be considered stable at given instances. If necessary, we can easily revert to these states – in case modification logs are preserved. The frequency of update sessions is closely linked to the consistency. When the desired consistency is weak and the goal is to obtain a maximum amount of information, it is advisable to reduce the time between sessions so that datasets can be updated as frequently as possible even if operations have to be often undone and redone. On the other hand, when strong consistency is desired for preparing future datasets, it makes sense to increase the time between update sessions so as to minimize operations on the data.

3.2 Sequencing of stages in the integration strategy

The first stage (verification of the relevance) takes place as soon as sets of evolutions are received (see Figure 2). The second and third phases take place only in mutual exclusivity. They can take place several times during the execution of the integration strategy's overall process. In fact, an update session can be triggered at any instant, for example, when all the proposed evolutions have been processed, or at periodic dates, pre-programmed before the start of the mission, or even at user request. In addition, the set of non-conflicting evolutions derived from the consistency checking phase serves as an entry point to the update session because it constitutes the set of evolutions to integrate in the user's dataset. The consistency checking stage should therefore be stopped as soon as an update session is triggered. As soon as the update session ends, consistency checking resumes if there are still remain evolutions to be processed or if new ones have arrived in the meantime.

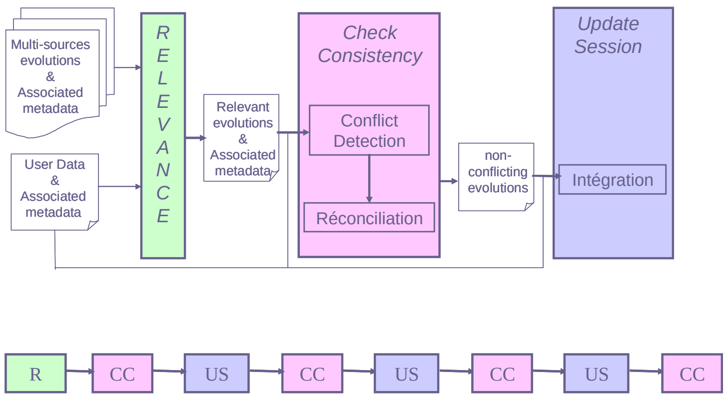

Figure 2: Sequencing of the integration strategy

4 Metadata model

We use metadata associated with the user, the data and the evolutions in two stage of the process: first, to filtering the data that are not suitable for the end user and second, to reconcile conflicting data which can provoke inconsistencies into the user dataset.

To supply shared and consistent knowledge of data between different communities, we have to standardize the metadata [17] [18] [19] [20]. Standards for describing metadata have been developed and provide a common base to the users [21] [22]. The use of normalized metadata in the infrastructure thus promotes interoperability between different actors and systems.

The standard that draws the most attention nowadays is [8]. The ISO 19115 standard defines a large set of metadata elements so that it can be used by several different types of users. However, a community of a given set of geographic-data users normally uses only a part of the different metadata elements defined in the standard and, in spite of the wide variety of elements, often needs to add elements not specified in the standard. ISO 19115 allows this thanks to the possibility of defining community profiles. A profile thus allows use of the standard restricted to a subset of mandatory elements and also extended by the addition of missing sections, entities and elements.

ISO 19115 has been designed to provide information on the use and exchange of datasets. The evolutions sets and basic evolutions (creation, deletion or modification) are not included in the standard. It is therefore not currently possible to provide metadata relating to a set of evolutions that we would want to provide to a user who already possesses the reference dataset.

Consequently, metadata elements have to be added to take these new requirements into account. We thus propose to extend ISO 19115 to encompass evolutions. Towards this end, we have created a metadata profile to manage evolutions.

Quality metadata defined by ISO 19115 are used to describe the data quality from the producer’s point of view, which is not sufficient for our context. We therefore propose to add quality elements to take the user’s point of view into account. We also restrict some elements of the standard which are not useful to us.

In ISO 19115, all the information on the quality is available thanks to a set of quality measurements accessible via the *DQElement* class of which only the name (attribute *nameOfMeasure*) and the result (attribute *result*) of quality measurements are specified in the profile (see figure 3).

We have then added a sub-class *MU_Usability*[3] to be able to judge the evolutions' ability to satisfy the usage the user may put them. This element indicates the degree of the evolutions' conformity with the usage that a given user type may put them to. This class has two attributes to indicate the type of user concerned by the quality measurement (attribute *finalUserRole*) and the site where the user is located (attribute *finalUserLocation*).

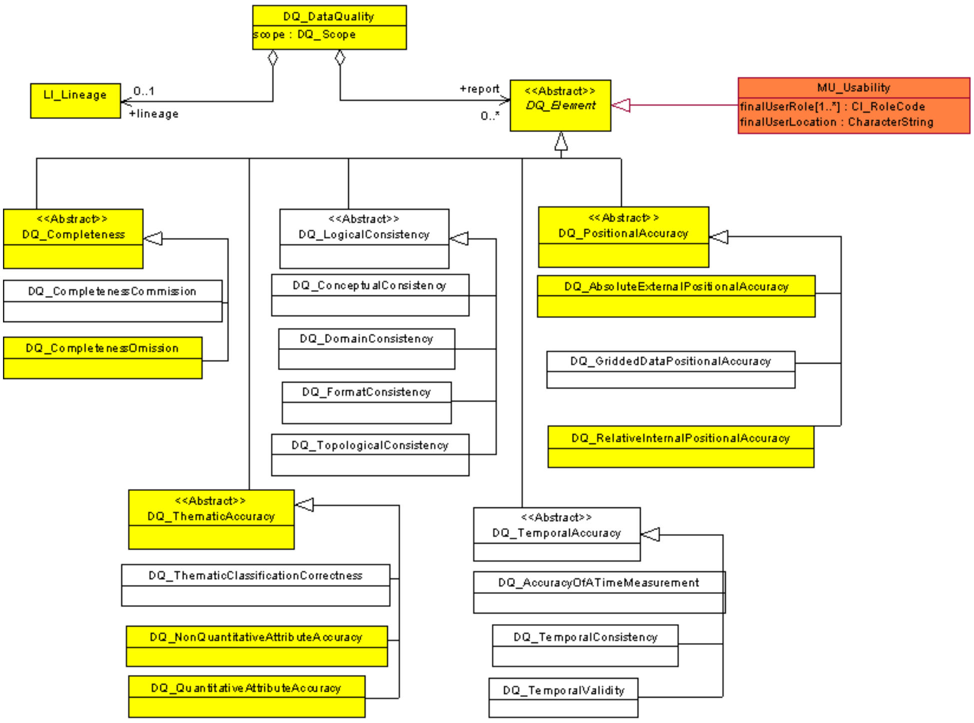

Figure 3: Figure 2: Quality elements in the profile

Finally, the expression of the result of the quality is shown in figure 4. The results available in the standard (*DQ_QuantitativeResult* and *DQ_ConformanceResult*) can be easily supplied by a producer but with difficulty, or not at all, by the other users. In fact, some users do not even have the technical means to evaluate the evolutions accurately and have to judge by themselves the quality of the updates they have made. We therefore

[3] The prefix MU_ is associated with the name of the profile in one context of work

think that the quantitative results are not sufficient to describe the information on the evolutions' quality and we have thus added qualitative elements to be able to judge the evolutions' quality in a more flexible manner. To this end, a class *MU_QualitativeResult* class has been created. It allows us, on the one hand, to quickly see whether non-spatial information attached to the basic evolutions has been correctly documented (attribute *documentation*) and, on the other, to see what are the types of errors that the updates may contain (attribute *errorType* of type *MUerror)*.

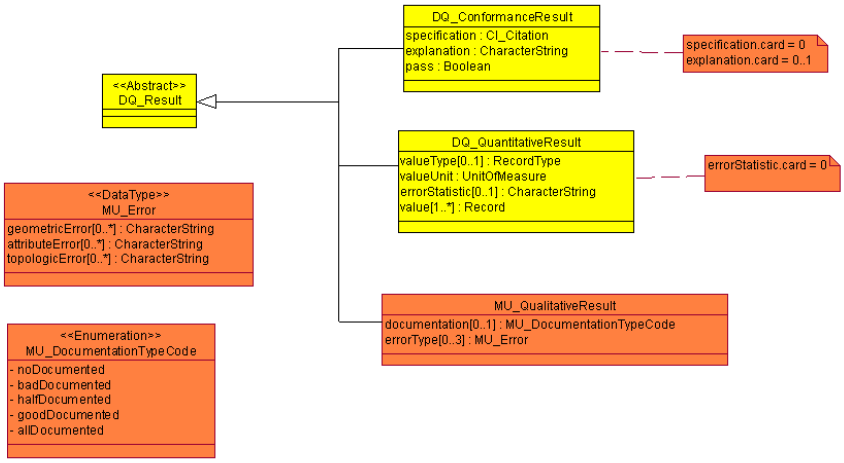

Figure 4 : Figure 3: Expression of the quality result in the profile

Infrastructure users do not all have the same requirements as far as evolutions are concerned. These requirements depend on several factors such as the user's role, the site where he is located, etc.

Indeed, a certain number of consistency constraints linked to the dataset used and to the technical methods available have to be defined for each actor. In fact, an infrastructure actor has a dataset that is, admittedly, derived from a unique reference set, but which may have been transformed so that it can be used by the user's system.

This entire information (needs and constraints) constitutes the actors' metadata. The needs are not fixed and can change over time. On the other hand, the constraints are imposed at the beginning and cannot be changed.

We have specified several criteria defining the entirety of user needs, such as the maximum spatial extent, the minimum occurrence date, the different thematic layers and the type of evolutions required. Also specified are the minimum geometric and semantic accuracies, as well as the reliability.

The list of consistency constraints has, on its part, been defined by spatial constraints (geometric accuracy, minimum resolution, etc.), semantic constraints (quantitative and qualitative accuracy) and context constraints (type of sources allowed, occurrence date and maximum extent of the dataset, etc.).

5 Validation/Results

The context of simulation that we have selected is that of an actor with a dataset derived from a reference set but which he has changed to suit his environment and his requirements. Evolutions originating from other infrastructure actors are proposed to our reference actor for possible integration. The baseline data used in our simulation environment are vector data format. Evolutions are furnished in an incremental format.

We have focused on the consistency checking part of the integration strategy. In particular, we have wanted to show that while the automatization of the process of concurrency control proves difficult in some circumstances, the case for the use of metadata for the reconciliation of conflicting data is convincing.
To validate the concurrency control, we have considered three evolution products. These products are derived from the updating of a reference dataset. In addition, we consider that the actor's dataset has already been updated by the actor himself, and thus differs from the reference dataset. For each product, we have undertaken automatic control and we have then verified it interactively. We have compared the results obtained using the mathematical concepts of precision[4] and recall[5].

Evolutions	Number of	Number of	Number of	Precision	Recall

[4] The precision is the number of correct results obtained divided by the total number of results.

[5] The recall is the number of correct results obtained divided by the desired number of correct results.

Product	evolutions	conflicts to detect	detected conflicts		
Product P1	14 evolutions	17	19	89%	100%
Product P2	21 evolutions	36	36	100%	100%
Product P3	41 evolutions	43	45	93%	98%

Tab 1 : Result of concurrency control

As anticipated, an analysis of the results shows that the use of identifiers greatly facilitates the concurrency control for those update conflicts that relate to modification or deletion types of evolutions (100% of conflicts detected for each product). However, this would require an effort to transform the datasets which do not yet use identifiers. We also observe that the automatic process for concurrency control that we have implemented detects some conflicts incorrectly (precision of 89% and 93% for two of the three products tested). In particular, we observe that the process detects a greater number of creation conflicts than actually exist and that these conflicts do not take place in reality. (Between evolutions of the creation type on the ‘infrastructure road network’ layer and the user data of product P3, 6 conflicts were detected – whereas only three should have been – and 3 of them proved incorrect.) This is due to the fact that the conflicts of creation are detected using a technique of geometric matching for which we have to define a threshold beyond which we consider the data matched. Depending on this value, we obtain more or fewer matched objects. The difficulty lies in defining this threshold value to obtain optimal results. Specifically, we have to find values that lead to a maximum number of matches without adding any false matches. To this end, we conducted several tests with different thresholds before obtaining matches which seemed to us to provide us the best results for our application context. The solution we finally retained leads to the detection of an acceptable number of real conflicts while limiting the number of incorrect matches. (The values for recall that we have obtained are always above 75%.) Nevertheless, in view of these results, we can state that the detection of conflicts using matching has proven the process most difficult to automatize.

As far as reconciliation is concerned, we have considered the situation in which some data and an evolution were declared to be in conflict and need to be processed. To be as close as possible to our study’s real context and to estimate as best as possible the reconciliation results, we have simulated the following two cases: in the first, the reference actor has requirements that can change over time. In the second case, the reference actor is a producer whose requirements were fixed at the beginning of the mission and cannot change. Proceeding in this manner, we can illustrate the reconciliation process as a function of each actor’s own constraints and as a function of the consistency level inherent to his role within the infrastructure. The process calculates a quality measurement, and then normalizes this measurement to obtain a result between -1 and 1. Closer the quality measurement is to -1, better is the external quality.

The overall quality measurements obtained from this simulation shows results that vary depending on the actor's role within the infrastructure (and thus the expected consistency level).

	Producer	Users
Evolution	-0.2246	-0.3118
Data	-0.437	-0.17

Tab 2 : Overall qualities measurements

In fact, we see in particular that the evolution will be more suited to the user's requirements whereas the data will suit the producer better. The process will therefore have a preference for one or the other conflicting data depending on the actor who will finally use the data after the integration. This result proves that it is possible to develop a reconciliation process which uses the information present in the metadata to be able to propose a choice between conflicting data. It does so by considering, on the one hand, the actor's requirements and constraints, established depending on his role within the infrastructure, and, on the other, the consistency levels desired for the different datasets to be used.

6 Conclusion

We have proposed a strategy for integrating evolutions that can manage a continuous stream of evolutions arriving from multiple sources in a context of decision making. This method, applicable to each site, relies on standardized metadata to arrive at the most suitable solution when confronted with a conflict and guarantees a result based on the desired level of consistency. This method thus promotes interoperability between different systems implemented and facilitates cooperation between actors.

Our strategy is divided into several stages, each designed to process a problem relating to the exchange and integration of multiple evolutions. To begin with, we deem it necessary to evaluate the relevance of the evolutions to filter out those that are not relevant to the end user's requirements. Then we use a process we have developed to check the consistency. This process consists of a concurrency control, which allows conflicting data to be detected, and of a reconciliation phase, which chooses which of two conflicting data is to be integrated. We show the utility of using normalized metadata during the reconciliation phase to be able to provide information necessary for making the best choice. This is done by considering, on the one hand, the user's requirements and constraints – as established by his role within the infrastructure – and on the other, the desired consistency levels for the different datasets to be used. Finally, in conjunction with

the consistency checking process, we have chosen to use update sessions to conduct a coherent integration of evolutions into the reference actor's dataset. In addition, we have shown the relative sequencing between these two final stages in the integration strategy.
One of the perspectives that we have been able to bring to this study is to continue to explore the aspect of the suitability to the user's requirements. This would allow us to further improve our results, both during the filtering out of irrelevant evolutions and during the reconciliation of conflicting data.

References

[1] D. Benslimane, F. Jouanot, R. Laurini, K. Yetongnon, N. Cullot & M. Savonnet: Interopérabilité des SIG: un état de l'art. Revue Internationale de géomatique, Volume 9 – no. 3/1999, pages 279 to 316.

[2] A. Braun: Conception d'outils d'aide à l'intégration des mises à jour dans les bases de données géographiques utilisateur. Etude interne IGN, (2003).

[3] P. Cederqvist, R. Pesch, et al., Version Management with CVS. (2001), http://www.cvshome.org/docs/manual.

[4] Esri, Versioning, An ESRI ® Technical Paper. (January 2004). www.esri.com

[5] Frank, A., Grum, E. and Vasseur, B. Procedure to select the best dataset for a task. In International Conference on Geographic Information Science. (2004).

[6] Grum, E. and Vasseur, B. How to select the best dataset for a task? In Fourth International Symposium on Spatial Data Quality ISSDQ'04, (2004), 197-206.

[7] Intergraph, White paper on Versioning, Lineage, Timestamps and Temporal Databases. Intergraph Corporation, (2003)

[8] Geographic Information – Metadata. ISO 19115:2003, International Organisation for Standardisation.

[9] Jeansoulin, R. and Wilson. Model-based semantics for ontologies of geographic information. In Gisciences, (2002)

[10] Kermarrec, A., Rowstron, A., Shapiro, M., and Druschel, P. The icecube approach to the reconciliation of divergent replicas. In Proceedings of the 20th ACM Symposium on Principles of Distributed Computing (PODC 2001), (26–29 August 2001). Newport, Rhode Island (USA).

[11] Nebert D. 2004. Developing Spatial Data Infrastructures: The SDI Cookbook, version 2.0.

[12] Oracle 2003, Oracle® Database Advanced Replication 10g Release 1 (10.1) Part No. B10732-01 December 2003.

[13] Rajabifard A. and Williamson I.P., Spatial Data Infrastructures: Concept, SDI Hierarchy and Future Directions. Proc. of GEOMATICS'80 Conference, Teheran, Iran (2001.).

[14] Y. Saito & M. Shapiro. Optimistic replication. ACM Computing Surveys, (2005), 37(1):42–81,

[15] Seshadri P., Garrett P. 2000. SQLServer for Windows CE - A Database Engine for Mobile and Embedded Platforms. Int. Conf. on Data Engineering (ICDE), (2000).

[16] Vasseur, B. Modélisation de l'information de qualité dans les applications Géographiques. Phd Thesis, Université Aix-Marseille 1, 2004.

[17] J. Barde, J. C. Desconnets, T. Libourel, P. Maurel (2006) Generic conceptual models for data and knowledge sharing. Application to environmental domain. Hydroscience and Engineering, ICHE 2006 16: 407-420

[18] Luzet, C. (1998). Megrin's gddd, moving to distributed metadata. In EOGEO98

[19] Gunther, O. et Voisard, A. (1997). Metadata in Geographical and Environmental Data Management. Managing Mutimedia Data : Using Metadata to Integrate and Apply Digital Data.

[20] Spéry, L. & Libourel, T. (1998). Vers une structuration des métadonnées. In Journées Cassini.

[21] CEN (1998). Geographic Information European Prestandards, Euronorme Voluntaire for Geographic Information -Data description- Metadata. European Committee for Standardization - CEN/TC287.

[22] FGDC (1998). Content Standard for Digital Geospatial Metadata, version 2.0. Document FGDC-SDT-001-1998. Federal Geographic Data Commitee, Metadata Ad Hoc Working Group]

.